

# CALL LOOP

## Structured Mode Syntax

```
CALL LOOP operand1 [operand2]...40
    statement...
END-LOOP
```

## Reporting Mode Syntax

```
CALL LOOP operand1 [operand2]...40
    statement...
[LOOP]
```

Operand	Possible Structure				Possible Formats												Referencing Permitted	Dynamic Definition
Operand1	C	S				A											yes	no
Operand2	C	S	A	G		A	N	P	I	F	B	D	T	L	C		yes	yes

## Function

The CALL LOOP statement is used to generate a processing loop that contains a call to a non-Natural program.

Unlike the CALL statement, the CALL LOOP statement results in a processing loop which is used to repeatedly call the non-Natural program. See the CALL statement for detailed description of CALL processing.

## Program Name - operand1

The name of the program to be called (*operand1*) can be specified as a constant or - if different programs are to be called dependent on program logic - as an alphanumeric variable of length 1 to 8. A program name must be placed left-justified in the variable.

## Parameters - operand2

The CALL LOOP statement can have a maximum of 40 parameters. The parameter list is constructed as described for the CALL statement. Fields used in the parameter list may be initially defined in the CALL LOOP statement itself or may have been previously defined.

## Loop Termination

The processing loop initiated with a CALL LOOP statement must be terminated with an ESCAPE statement.

## Restriction

The statements AT BREAK, AT START OF DATA and AT END OF DATA must not be used within a CALL LOOP processing loop.

## Example

```
DEFINE DATA LOCAL
1 PARAMETER1 (A10)
END-DEFINE
CALL LOOP 'ABC' PARAMETER1
  IF PARAMETER1 = 'END'
    ESCAPE BOTTOM
  END-IF
END-LOOP
END
```